PORTAL
THE ACM DIGITAL LIBRARY

> home > about > feedback > login

US Patent & Trademark Office

Try the *new* Portal design
Give us your opinion after using it.

Search Results

Search Results for: **[kron<AND>((transformational AND grammars AND languages AND compilers))]**
Found **15** of **127,944** searched.

Best Available Copy

Search within Results

> Advanced Search

> Search Help/Tips

Sort by: Title Publication Publication Date Score Binder

Results 1 - 15 of 15 short listing

1 Table compression for tree automata 85%
Jürgen Börstler , Ulrich M[illegible]nhard Wilhelm
ACM Transactions on Prog[illegible]ng Languages and **Systems (TOPLAS)** July 1991
Volume 13 Issue 3

2 Unified versioning through feature logic 82%
Andreas Zeller , Gregor Snelting
ACM Transactions on Software Engineering and Methodology **(TOSEM)** October 1997
Volume 6 Issue 4
Software configuration management (SCM) suffers from tight coupling between SCM version-ing models and the imposed SCM processes. In order to adapt SCM tools to SCM processes, rather than vice versa, we propose a unified versioning model, the version set model. Version sets denote versions, components, and configurations by feature terms, that is, [illegible] terms over (feature : value)-attributions. Through feature logic, we ...

3 Extending attribute grammars to support programming-in-the-large 82%
Josephine Micallef , Gail E. Kaiser
ACM Transactions on Programming Languages and Systems **(TOPLAS)** September 1994
Volume 16 Issue 5
Attribute grammars add specification of static semantic properties to context-free grammars, which, in turn, describe the syntactic structure of program units. However, context-free grammars cannot express programming-in-the-large features common in modern programming languages, including unordered collections of units, included units, and sharing of included units. We present extensions to context-free grammars, and corresponding extensions to attribute grammars, suitable for defining such ...

4 A truly generative semantics-directed compiler generator 80%

Harald Ganzinger , Robert Giegerich , Ulrich Möncke , Reinhard Wilhelm
ACM SIGPLAN Notices , Proceedings of the 1982 SIGPLAN symposium on Compiler construction June 1982
Volume 17 Issue 6

This paper describes semantic processing in the compiler generating system MUG2. MUG2 accepts high-level descriptions of the semantics of a programming language including full runtime semantics, data flow analysis, and optimizing transformations. This distinguishes MUG2 from systems such as YACC [Joh75], HLP [HLP78], PQCC [PQC79], or its own former version [GRW77] with respect to expressive power and convenience. In this respect MUG2 comes close to semantics-directed systems such as [Mos76 ...

5 Code generation using tree matching and dynamic programming 80%

Alfred V. Aho , Mahadevan Ganapathi , Steven W. K. Tjiang
ACM Transactions on Programming Languages and Systems (TOPLAS) October 1989
Volume 11 Issue 4

Compiler-component generators, such as lexical analyzer generators and parser generators, have long been used to facilitate the construction of compilers. A tree-manipulation language called twig has been developed to help construct efficient code generators. Twig transforms a tree-translation scheme into a code generator that combines a fast top-down tree pattern matching algorithm with dynamic programming. Twig has been used to specify an ...

6 Tree transformation techniques and experiences 80%

S. E. Keller , J. A. Perkins , T. F. Payton , S. P. Mardinly
ACM SIGPLAN Notices , Proceedings of the 1984 SIGPLAN symposium on Compiler construction June 1984
Volume 19 Issue 6

A formal description technique for describing transformations from one well-defined language to another is introduced. A TT grammar contains context-free grammars for describing the syntax of both languages. The transformation between the languages is described by a relation on the productions from the grammars. The TT-grammar is supported by an automatic tool, SSAGS -- a translator writing system based on attribute grammars -- which will support certain classes of TT-grammars. SSAGS a ...

7 Pattern Matching in Trees 80%

Christoph M. Hoffmann , Michael J. O'Donnell
Journal of the ACM (JACM) January 1982
Volume 29 Issue 1

8 An improvement to bottom-up tree pattern matching 80%

D. R. Chase
Proceedings of the 14th ACM SIGACT-SIGPLAN symposium on Principles of programming languages January 1987

9 A unified version model for configuration management 80%

Andreas Zeller
ACM SIGSOFT Software Engineering Notes , Proceedings of the 3rd ACM SIGSOFT

symposium on Foundations of software engineering October 1995
Volume 20 Issue 4

10 Software reuse 80%
Charles W. Krueger
ACM Computing Surveys (CSUR) June 1992
Volume 24 Issue 2
Software reuse is the process of creating software systems from existing software rather than building [illegible] This simple yet powerful vision was introduced in 19[illegible], however, failed to become a standard software engineering [illegible] attempt to understand why, researchers have renewed their interest in software reuse and in the obstacles to implementing it. This paper surveys the different approaches to software reuse found in the ...

11 Version control in [illegible] 77%
J. F. H. Winkler
Proceedings of the 9th international conference on **Software Engineering** March 1987
Programs products are quite often families of large and modular programs. Modern programming languages support the family [illegible] such program families only partially. At the time [illegible] describe different revisions, variants, and versions [illegible] cks and whole programs. This paper presents a pr[illegible] version information as part of the program text. In [illegible] part of a program building block th ...

12 Techniques and langu[illegible]ping generic informations systems: a case study 77%
Silke Eckstein , Peter Ahlbrecht , Karl Neumann
ACM SIGSOFT Software Engineering Notes , **Proceedings of the 2001 symposium on Software reusability**: putting software reuse in context May 2001
Volume 26 Issue 3
When creating a family of [illegible] several systems of similar type which differ within some aspects, it [illegible] express these differences already at the level of the spec[illegible] tain systems from it which are ready to run. The use of [illegible] may lead to substantial progress in this area. This report explore[illegible] parameterization concepts at the specification level, which can be used [illegible] variant's of a system, and gene ...

13 Developing laborator[illegible]ting laboratory repository: guidelines, [illegible] sample labs 77%
Daniel Joyce , Deborah K[illegible] , Cary Laxer , Kenneth Loose , [illegible] Whitehurst
ACM SIGCUE Outlook October 1997
Volume 25 Issue 4

14 Developing laborat[illegible]ting laboratory repository: guidelin[illegible] sample labs (report of the ITiCSE '97 working gr[illegible]ory materials for computing courses) 77%
Daniel Joyce , Deborah K[illegible] Koffman , Wolfgang Kreuzer ,

Cary Laxer , Kenneth Loose , Eric Sutherland , R. Mark Whitehurst
The supplemental proceedings of the conference on Integrating technology into computer science education: working group reports and supplemental proceedings June 1997

15 Reengineering of conceptual models from mathematical concept analysis 77%
Gregor Snelting
ACM Transactions on Software Engineering and Methodology **(TOSEM)** April 1996
Volume 5 Issue 2

Results 1 - 15 of 15 show history

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2004 ACM, Inc.

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☑ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.